

功能：计算B样条曲线 $q(u)$ 到B样条曲线 $p(u)$ 的有向豪斯道夫距离Delta及其在两曲线上的对应点 $(qx, qy)$ 与 $(px, py)$ 。  
 输入参数： $[a, b]$ —两曲线 $q(u)$ 与 $p(u)$ 的定义区间；B样条曲线 $p(u)$ 由控制顶点 $m\_aVertex$ 与节点矢量 $m\_aNode$ 定义，B样条曲线 $q(u)$ 由控制顶点 $m\_newVertex$ 与节点矢量 $m\_newNode$ 定义，都是受保护成员。  
 输出参数：Delta—B样条曲线 $q(u)$ 到 $p(u)$ 的有向豪斯道夫距离， $(qx, qy)$ 与 $(px, py)$ —分别是两曲线上的对应点。

```
void CuvQtoPHausdorff(double a, double b, double &Delta0,
                      double &qx, double &qy, double &px, double &py)
{
    //B样条曲线p(u)的控制顶点为CArray<CPoint, CPoint> m_aVertex,
    //节点矢量为CArray<double, double> m_aNode, 都属受保护成员。
    //B样条曲线q(u)控制顶点为CArray<CPoint, CPoint> m_newVertex;
    //节点矢量为CArray<double, double> m_newNode, 都属受保护成员。
    CArray<double, double> p0, p1, maxq, maxp;
    maxq.SetSize(2); maxp.SetSize(2);
    double u0=a;
    double u1=b;
    double xq=0., yq=0.;
    double eps=0.000001;
    double d0, d1, d_l, u_l, uq, up0, up1;
    int k=m_nTimes; /* m_nTimes现在是降阶后次数 */
    int imax;
    double maxd=0.;
    double ratio=(sqrt(5.)-1.)/2.; /* 黄金分割数0.618...的精确值 */
    double uh=(b-a)/10.;
    for(int i=0; i<=10; i++) /* 将[a, b]等分成10个子区间, 找出11个
                                分点中的最大偏差点imax */
    {
        u0=a+i*uh;
        GetBQr(0, k, u0, qx, qy);
        uq=u0;
        QtoPCuvDistance(k, k+1, uq, a, b, d0, qx, qy, px, py);
        if(d0>=maxd) {maxd=d0; imax=i;}
    }
    if(imax>0&&imax<10)
    {
        u_l=a+(imax-1)*uh; /* 确定最大偏差点imax前后子区间中哪一个是
        最大偏差, 用左端点号imax标记该子区间 */
        GetBQr(0, k, u_l, qx, qy);
        uq=u_l;
        QtoPCuvDistance(k, k+1, uq, a, b, d_l, qx, qy, px, py);
        u1=a+(imax+1)*uh;
        GetBQr(0, k, u1, qx, qy);
        uq=u1;
        QtoPCuvDistance(k, k+1, uq, a, b, d1, qx, qy, px, py);
        if(d_l>=d1) {maxd=d0; imax=imax-1;}
    }
    if(imax==10) imax=9;
    u0=a+imax*uh;
    GetBQr(0, k, u0, qx, qy);
    uq=u0;
    QtoPCuvDistance(k, k+1, uq, a, b, d0, qx, qy, px, py);
    if(d0>=maxd) {maxd=d0; maxq[0]=qx; maxq[1]=qy; maxp[0]=px; maxp[1]=py;}
    u1=a+(imax+1)*uh;
    GetBQr(0, k, u1, qx, qy);
    uq=u1;
    QtoPCuvDistance(k, k+1, uq, a, b, d1, qx, qy, px, py);
    if(d1>=maxd) {maxd=d1; maxq[0]=qx; maxq[1]=qy; maxp[0]=px; maxp[1]=py;}
    int nn=0; /* 进入while循环执行迭代次数统计 */
    while(fabs(u1-u0)>0.000000000001) /* 对最大偏差区间用黄金分割法精搜索
                                最大曲线偏差 */
    {
        nn=nn+1;
        if(d0>=d1)
        {
            u1=u0+ratio*(u1-u0);
            GetBQr(0, k, u1, qx, qy);
            up1=u1;
            QtoPCuvDistance(k, k+1, up1, a, b, d1, qx, qy, px, py);
            if(d1>=maxd&&up1>=a&&up1<=b)
                {maxd=d1; maxq[0]=qx; maxq[1]=qy; maxp[0]=px; maxp[1]=py;}
        }
        if(d0<d1)
        {
            u0=u0+(1-ratio)*(u1-u0);
            GetBQr(0, k, u0, qx, qy);
            up0=u0;
        }
    }
}
```

```

                                CuvQtoPHausdorff.txt
QtoPCuvDistance(k,k+1,up0,a,b,d0,qx,qy,px,py);
if(d0>=maxd&&up0>=a&&up0<=b)
    {maxd=d0; maxq[0]=qx; maxq[1]=qy; maxp[0]=px; maxp[1]=py;}
}
if(fabs(d1-d0)<=0.0000000001)
{
    Delta0=maxd;
    qx=maxq[0];  qy=maxq[1];
    px=maxp[0];  py=maxp[1];
    return;
}
}
Delta0=maxd;
}

```